

defined as a software component) at the same time, where exclusive access to the object is required by one or both of these threads. Such concurrent access of the same object may result in corruption of programs' data structures, ultimately causing the computer to crash. Therefore, when a given thread accesses an object, generally it is provided a lock on that object, for example, utilizing a lock object so that other threads can only acquire limited rights to the object until the given thread is finished with using the object. For instance, to improve thread throughput, it may be desirable to implement Read-Write locks, allowing for multiple threads to read the data structure, but only one thread modifying it. Another example may be a SQL database, in which appending is a very common situation, and thus it is desirable to make it a non-blocking procedure, which is often done by implementing a complex lock structure.

Please amend the paragraph beginning at page 12, line 18, as follows:

The recyclable locking mechanism 206 associates a lock within the pool 204, such as the lock object 208, with an object, such as the object 202, using the associated variable of the object as a pointer to the lock, upon a first request by a thread, such as the thread 200, in one embodiment where the object 202 has not been previously locked by another thread. Pursuant to a second request by the thread, the mechanism 206 further deassociates the lock from the object, such that the lock is recycled, that is, can be reused.

Please amend the paragraph beginning at page 15, line 12, as follows:

Referring now to FIG. 3, a flowchart of a method according to one embodiment of the invention is shown. In 300, which is an initialization phase, an associated variable of an object, such as the object 202 of FIG. 2, is reset by in one embodiment by being set to -1. The associated variable includes a set of high bits and a set of low bits, where the set of high bits followed by a zeroed set of low bits acts as a lock pointer, and the set of low bits acts as a status variable regarding whether the object is currently being used. The number of low bits that

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{LLC}
1420 Fifth Avenue, Suite 2800
Seattle, Washington 98101
206.682.8100

AL
corresponds has to be greater or equal to the logarithm with base 2 of the maximum number of threads they may attempt to associate a lock with an object at the same time. Moreover, it is assumed that all locks are aligned at addresses of which all low bits are zero.

Please amend the paragraph beginning at page 16, line 22, as follows:

If the associated variable is less than the boundary value, but nonzero, this signifies that none of the high bits have been set to one, such that the set of high bits followed by a number of zero bits equal to the number of low bits (that is, the associated variable with the low bits zeroed) does not point to a lock object. Thus, the object has a spin status, meaning that a lock object is in process of being assigned to the object, such that in 309 the thread waits until the lock object is pointed to by the set of high bits of the associated variable of the object, as the set of high bits is followed by a number of zero bits equal to the number of low bits. In this case, the thread shall yield control, for instance by calling Win32 function Sleep().

Please amend the paragraph beginning at page 18, line 8, as follows:

Once the thread has finished using the object, it decrements the associated variable of the object, in one embodiment by 1, in 314. In another embodiment, the thread accomplishes this decrementation by sending a second, unlock request to the recyclable locking mechanism, and the mechanism itself performs the decrementation. In another embodiment, the thread accomplishes the decrementation itself. In 316, it is determined whether the associated variable is equal to a predetermined threshold, in one embodiment, -1. If this is the case, then this means that no other threads are desiring to use this object (i.e., no other objects are waiting in 309 in their traversal through the flowchart of FIG. 3), and the lock object is recycled in 318 for reuse; the method then ends at 320. If this is not the case, then the method ends at 320 without recycling the lock object, since it is still being used to lock the object because the object is being used-by other threads.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLC}
1420 Fifth Avenue, Suite 2800
Seattle, Washington 98101
206.682.8100